[MS-SPS2SAUTH]:

OAuth 2.0 Authentication Protocol: SharePoint Profile

Revision Summary

Date	Revision History	Revision Class	Comments
1/20/2012	0.1	New	Released new document.
4/11/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	1.1	Minor	Clarified the meaning of the technical content.
7/30/2013	1.1	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.2	Minor	Clarified the meaning of the technical content.
7/31/2014	1.2	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.2	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	2.0	Major	Significantly changed the technical content.
7/15/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/24/2018	2.0	None	No changes to the meaning, language, or formatting of the technical content.
10/1/2018	2.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The OAuth 2.0 Authentication Protocol: SharePoint Profile is used for server-to-server authentication between server-side applications.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

authentication: The act of proving an identity to a server while providing key material that binds the identity to subsequent communications.

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [RFC7159]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

principal: An authenticated entity that initiates a message or channel in a distributed system.

realm: (1) An administrative boundary that uses one set of authentication servers to manage and deploy a single set of unique identifiers. A realm is a unique logon space.

(2) A collection of key distribution centers (KDCs) with a common set of principals, as described in [RFC4120] section 1.2.

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL supports server and, optionally, client **authentication** using X.509 certificates [X509] and [RFC5280]. SSL is superseded by **Transport Layer Security (TLS)**. TLS version 1.0 is based on SSL version 3.0 [SSL3].

Security Assertion Markup Language (SAML): The set of specifications that describe security assertions encoded in XML, profiles for attaching assertions to protocols and frameworks,

request/response protocols used to obtain assertions, and the protocol bindings to transfer protocols, such as SOAP and HTTP.

security principal: A unique entity that is identifiable through cryptographic means by at least one key. It frequently corresponds to a human user, but also can be a service that offers a resource to other security principals. Also referred to as principal.

security principal identifier: A value that is used to uniquely identify a security principal. In Windows-based systems, it is a security identifier (SID). In other types of systems, it can be a user identifier or other type of information that is associated with a security principal.

security token: An opaque message or data packet produced by a Generic Security Services (GSS)-style **authentication** package and carried by the application protocol. The application has no visibility into the contents of the token.

security token service (STS): A web service that issues claims and packages them in encrypted security tokens.

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

site: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. **TLS** supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). **TLS** is standardized in the IETF TLS working group.

user principal name (UPN): A user account name (sometimes referred to as the user logon name) and a domain name that identifies the domain in which the user account is located. This is the standard usage for logging on to a Windows domain. The format is: someone@example.com (in the form of an email address). In Active Directory, the userPrincipalName attribute of the account object, as described in [MS-ADTS].

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[IETFDRAFT-JWT-LATEST] Jones, M., Bradley, J., and Sakimura, N., "JSON Web Token (JWT) draft-ietf-oauth-json-web-token-08", draft-ietf-oauth-json-web-token-08, May 2013, http://datatracker.ietf.org/doc/draft-ietf-oauth-json-web-token/

[IETFDRAFT-JWTOAuth] Jones, M., Campbell, B., and Mortimore, C., "JSON Web Token (JWT) Bearer Token Profiles for OAuth 2.0", July 2012, http://tools.ietf.org/html/draft-ietf-oauth-jwt-bearer-01

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-OAUTH2EX] Microsoft Corporation, "OAuth 2.0 Authentication Protocol Extensions".

[MS-ODATA] Microsoft Corporation, "Open Data Protocol (OData)".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

[RFC793] Postel, J., Ed., "Transmission Control Protocol: DARPA Internet Program Protocol Specification", RFC 793, September 1981, http://www.rfc-editor.org/rfc/rfc793.txt

1.2.2 Informative References

[IETFDRAFT-OAuth2.0] Hammer-Lahav, E., Ed., Recordon, D., and Hardt, D., "The OAuth 2.0 Authorization Protocol", draft-ietf-oauth-v2-22, http://tools.ietf.org/html/draft-ietf-oauth-v2-23

[MS-SPSTWS] Microsoft Corporation, "SharePoint Security Token Service Web Service Protocol".

[MS-XOAUTH] Microsoft Corporation, "OAuth 2.0 Authorization Protocol Extensions".

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.org/rfc/rfc2616.txt

1.3 Overview

This protocol specifies the profile of server-to-server **authentication** performed by an application server and other server-side applications such as a mail server. An example scenario is where an application server calls to a mail server to request access to tasks assigned to a user. The communication between the application server and mail server will use this protocol.

1.4 Relationship to Other Protocols

This protocol relies on the OAuth 2.0 Authentication Protocol Extensions, as described in [MS-OAUTH2EX], and JSON Web Token (JWT), as described in [IETFDRAFT-JWT-LATEST]. This protocol is related to the OAuth 2.0 Authorization Protocol Extensions as described in [MS-XOAUTH] that also rely on [MS-OAUTH2EX] for similar server-to-server scenarios.

This protocol uses HTTP, as described in [RFC2616], and HTTPS, as described in [RFC2818], as shown in the following layering diagram.

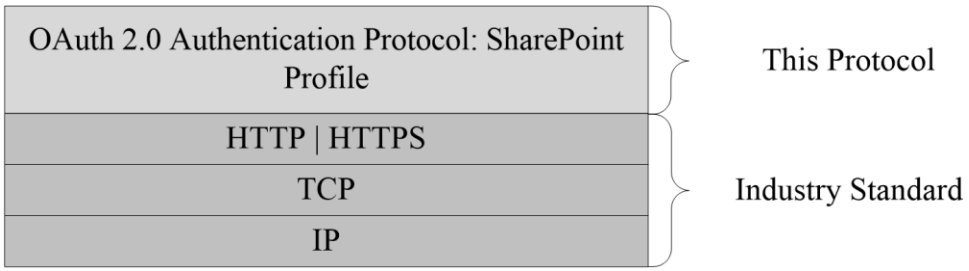

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol requires that the caller that is requesting a server-to-server token resides in the same system as **STS** as described by [MS-SPSTWS].

1.6 Applicability Statement

This protocol is designed for use by server-side applications that need to access protected resources and will use server-to-server **authentication**.

This protocol is intended only to be used over RESTful service calls.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol transports messages over **TCP**, as specified in [RFC793], and does not pass any specific parameters to the transport. This protocol uses **HTTPS**, as specified in [RFC2818], to secure the **security tokens**.

Messages are not encoded by the OData protocol, as specified in [MS-ODATA]. Messages use the default character set defined by the protocol client or the protocol server.

2.2 Message Syntax

A **security principal** is represented as a **security principal identifier** in the applications. A security principal identifier is a **GUID**. The following security principal identifier values are reserved values, but not the complete set of possible values, that are used in security tokens described throughout this document:

- 00000003-0000-0ff1-ce00-000000000000<1>
- 00000002-0000-0ff1-ce00-000000000000<2>
- 00000004-0000-0ff1-ce00-000000000000<3>

3 Protocol Details

3.1 Application Server Acting as Server Role Details

The application server<4> is in a resource server role and grants access to a protected resource. Another server, such as a mail server, is in a client role and makes protected resource requests. For clarity, the server making requests is referred to as the client application throughout this section and subsections.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The following sequence of events occurs for the client application to authenticate with the application server.

Step 1: The client application makes an anonymous service call to the application server.

Step 2: The application server returns an **HTTP** 401 challenge with an empty **Bearer** authorization header. The **Bearer** authorization header is specified in [IETFDRAFT-JWTOAuth].

The response contains the following parameters:

- *client_id*: An application identifier. The value MUST be 00000003-0000-0ff1-ce00-000000000000.
- *realm*: The source **realm (2)** of the application. The format of realm is specified in [MS-OAUTH2EX].
- *trustedissuers*: The list of the name identifiers of the issuers that the application server trusts.

Step 3: The client application creates a server-to-server token that contains the user identity information as an outer token. The following table describes claims that are used in the outer token, and are exchanged in server-to-server **security tokens**. The claim values are all string data types, as specified in [MS-DTYP]. All values in any server-to-server tokens MUST be lowercase strings.

Claim type	Claim description	Required value formats
aud	The audience that is the targeted service for which the token is issued. This claim type MUST be provided.	The value MUST be specified in the following format, where *hostname* is the application server's host name, and *realm* is the realm (2) provided in the HTTP 401 response.

Claim type	Claim description	Required value formats
		00000003-0000-0ff1-ce00-000000000000/*hostname*@*realm*
iss	The **principal** of the issuer. This claim type MUST be provided.	Any string format is allowed. The following format is typical, where *principalconfiguredguid* is preferably a **GUID**, but it can also be a name. *principalid@principalconfiguredguid*
nameid	The name identifier that is the value of the principal that makes the request, such as the signed-in user's UPN value.	Any string format is allowed. In general the following format is a typical format. *domain\user*
nii	The name identifier issuer.	If the name identifier was issued with **identityprovider** equal to "windows", then the following string is used. urn:office:idp:activedirectory If the name identifier was issued by custom forms-based membership providers, then the following format is used, where *membershipprovidername* is the name of the membership provider. urn:office:idp:forms:*membershipprovidername* If the name identifier was issued by a **SAML** identity provider, then the following format is used, where *samlprovidername* is the name of the SAML provider. urn:office:idp:trusted:*samlprovidername*
nbf	The *not_before* time at which the token was created. This claim type MUST be provided.	The format of this value is specified in [MS-OAUTH2EX] section 3.1.1.
exp	The *expires_on* time at which the token expires. This claim type MUST be provided.	The format of this value is specified in [MS-OAUTH2EX] section 3.1.1.
trustedfordelegation	A value indicating whether the caller is trusted to delegate a user identity.	The value MUST be one of the following values: ▪ true ▪ false
identityprovider	A value indicating the identity provider who authenticated the caller.	The value MUST be one of the following values: ▪ windows ▪ accesstoken ▪ forms ▪ trusted
actortoken	A value that points to the security token issued and signed by a trusted	The value is an application identity token

Claim type	Claim description	Required value formats
	issuer.	described in the next claims table.
smtp	The logged-on user's email address. This is an additional claim that trusted issuers send.	Any string format is allowed. For example, user@contoso.com.
sip	The logged on user's **SIP** address. This is an additional claim that trusted issuers send.	Any string format is allowed. The claim value depends on what is configured as the SIP address for the user. For example, sip:user@contoso.com.

Step 4: The client application constructs an application identity token which is inserted into the outer token as the value of the **actortoken** claim. The following table describes claims that are used in the application identity token. The claim values are all string data types, as specified in [MS-DTYP]. All values in any server-to-server tokens MUST be lowercase strings.

Claim type	Claim description	Required value formats
aud	The audience that is the targeted service for which the token is issued. This claim type MUST be provided.	The value MUST be specified in the following format, where *hostname* is the application server's host name, and *realm* is the realm (2) provided in the HTTP 401 response. 00000003-0000-0ff1-ce00-000000000000/*hostname*@*realm*
iss	The principal of the issuer. This claim type MUST be provided.	Any string format is allowed. The following format is typical, where *principalconfiguredguid* is preferably a GUID, but it can also be a name. *principalid@principalconfiguredguid*
nameid	The name identifier that is the value of the principal that makes the request, such as the signed-in user's UPN value.	The value MUST use the following format where *realm* is the realm (2) provided in the HTTP 401 response. 00000003-0000-0ff1-ce00-000000000000@*realm*
nii	The name identifier issuer.	If the name identifier was issued with **identityprovider** equal to "windows", then the following string is used. urn:office:idp:activedirectory If the name identifier was issued by custom forms-based membership providers, then the following format is used, where *membershipprovidername* is the name of the membership provider. urn:office:idp:forms:*membershipprovidername*. If the name identifier was issued by a SAML identity provider, then the following format is used, where *samlprovidername* is the name of the SAML provider. urn:office:idp:trusted:*samlprovidername*
nbf	The *not_before* time at which the token was created. This claim type MUST be provided.	The format of this value is specified in [MS-OAUTH2EX] section 3.1.1.

Claim type	Claim description	Required value formats
exp	The *expires_on* time at which the token expires. This claim type MUST be provided.	The format of this value is specified in [MS-OAUTH2EX] section 3.1.1.
trustedfordelegation	A value indicating whether the caller is trusted to delegate a user identity.	The value MUST be one of the following values: ▪ true ▪ false
identityprovider	A value indicating the identity provider who authenticated the caller.	The value MUST be one of the following values: ▪ windows ▪ forms ▪ trusted

Step 5: The client application sends the server-to-server token, which includes the outer token with user identity information, to the application server. The server-to-server token MUST be compatible with the **JSON** web token format specified in [IETFDRAFT-JWT-LATEST] and [MS-OAUTH2EX].

Step 6: The application server validates the server-to-server token and extracts the user identity information.

A relying party application accepts server-to-server tokens as long as the following criteria are met:

- The token is signed with one of the application server's trusted signing certificates.
- The token contains at least one of the following claims:
  - **nid** claim with the **UPN** value
  - **smtp** claim
  - **sip** claim
- The **iss** claim value in the outer token matches the **nameid** claim value in the inner token. The match is case sensitive.
- The **aud** claim value passes the audience validation check, which includes the following:
  - The **aud** claim MUST contain these parameters: *client_id*, *hostname*, and *realm*. The match is case sensitive.
  - The *client_id* parameter MUST be 00000003-0000-0ff1-ce00-000000000000.
  - The *hostname* parameter is the host name of the application server's **endpoint**.
  - The *realm* parameter matches the requested resource's realm (2).

The application server uses the claims in the token to grant access to its resources based on the user profile.

This protocol is used for the following endpoints on the application server:

- Client.svc
- Listdata.svc
- Sites.asmx
- _api

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Application Server Acting as Client Role Details

The application server<5> acts in a client role and makes protected resource requests to another server, such as a mail server, that grants access to a protected resource.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The following sequence of events occurs for the application server to authenticate with the server.

Step 1: The application server makes an anonymous service call to the relying party service. The call contains an empty value in the **Bearer** authentication scheme as specified in [IETFDRAFT-JWTOAuth].

Step 2: The relying party service returns an **HTTP** 401 challenge.

This response contains the following optional parameters:

- *client_id*: The application server MUST use the value 00000003-0000-0ff1-ce00-000000000000, which is the application identifier.
- *realm*: The **realm (2)** of the application **endpoint**. The format of realm is specified in [MS-OAUTH2EX].

- *trustedissuers*: The comma-separated list of the name identifiers of the issuers that the relying party application trusts.

Step 3: The application server adds the currently logged-on user's identity information as an outer token to the server-to-server token. This allows the application to convey the user information to the relying party service. The following table describes claims that are used in the outer token. The claim values are all string data types, as specified in [MS-DTYP]. All values in any server-to-server tokens MUST be lowercase strings.

Claim type	Claim description	Required value formats
aud	The audience that is the targeted service for which the token is issued. This claim type MUST be provided.	The value MUST use one of the following **security principal identifiers**: ▪ 00000002-0000-0ff1-ce00-000000000000 ▪ 00000004-0000-0ff1-ce00-000000000000 The value MUST be specified in the following format, where *principalid* is one of the previous security principal identifiers, *hostname* is the application server's host name, and *realm* is the realm (2) provided in the HTTP 401 response. *principalid*/*hostname*@*realm*
iss	The **principal** of the issuer. This claim type MUST be provided.	The value MUST use the following format, where *realm* is the realm (2) provided in the HTTP 401 response. 00000003-0000-0ff1-ce00-000000000000@*realm*
nid	The name identifier that is the logged-on user's UPN value of the principal that makes the request.	Any string format is allowed. In general the following format is a typical format. *domain**user*
identityprovider	String value indicating the identity provider who authenticated the caller. This is an additional claim that the **site** server issues and not required by OAuth 2.0 Authentication Protocol Extensions [MS-OAUTH2EX].	The value MUST be one of the following values: ▪ windows ▪ forms ▪ trusted
smtp	The logged-on user's email address. This is an additional claim that trusted issuers send.	Any string format is allowed. For example, user@contoso.com.
actortoken	A value that points to the security token issued and signed by a trusted issuer.	The value is an application identity token described in the next claims table.

Step 4: The application server constructs an application identity token which is inserted into the outer token as the value of the **actortoken** claim. The following table describes claims that are used in the application identity token. The claim values are all string data types, as specified in [MS-DTYP]. All values in any server-to-server tokens MUST be lowercase strings.

Claim type	Claim description	Required value formats

Claim type	Claim description	Required value formats
aud	The audience that is the targeted service for which the token is issued. This claim type MUST be provided.	The value MUST use one of the following security principal identifiers: ▪ 00000002-0000-0ff1-ce00-000000000000 ▪ 00000004-0000-0ff1-ce00-000000000000 The value MUST be specified in the following format, where *principalid* is one of the previous security principal identifiers, *hostname* is the application server's host name, and *realm* is the realm (2) provided in the HTTP 401 response. *principalid*/*hostname*@*realm*
iss	The principal of the issuer. This claim type MUST be provided.	Any string format is allowed. The following format is typical, where *principalconfiguredguid* is preferably a **GUID**, but it can also be a name. *principalid@principalconfiguredguid*
nameid	The name identifier that is the value of the principal that makes the request, such as the signed-in user's UPN value.	The value MUST use the following format, where *realm* is the realm (2) provided in the HTTP 401 response. 00000003-0000-0ff1-ce00-000000000000@*realm*
nii	The name identifier issuer.	If the name identifier was issued with **identityprovider** equal to "windows", then the following string is used. urn:office:idp:activedirectory If the name identifier was issued by custom forms-based membership providers, then the following format is used, where *membershipprovidername* is the name of the membership provider. urn:office:idp:forms:*membershipprovidername*. If the name identifier was issued by a **SAML** identity provider, then the following format is used, where *samlprovidername* is the name of the SAML provider. urn:office:idp:trusted:*samlprovidername*
nbf	The *not_before* time at which the token was created. This claim type MUST be provided.	The format of this value is specified in [MS-OAUTH2EX] section 3.1.1.
exp	The *expires_on* time at which the token expires. This claim type MUST be provided.	The format of this value is specified in [MS-OAUTH2EX] section 3.1.1.
trustedfordelegation	A value indicating whether the caller is trusted to delegate a user identity.	The value MUST be one of the following values: ▪ true

Claim type	Claim description	Required value formats
		▪ false
identityprovider	A value indicating the identity provider who authenticated the caller.	The value MUST be one of the following values: ▪ windows ▪ forms ▪ trusted

Step 5: The application server sends the server-to-server token, which has additional user information, to the relying party service.

Step 6: The relying party service validates the server-to-server token and extracts the user identity information.

Serialized user information

The application server accepts serialized user information that is a **JSON**-encoded key-value pair, similar to the following example, in order to make an outgoing server-to-server call.

{"typ":1,"idk":"bmFtZWlkDQpkdGF5bG9yQG1pY3Jvc29mdC5jb20NCg==","idp":"windows"}

- **typ** is the token type, where the value of 1 indicates that the information is for an application and user identities. A value of 2 indicates that the information is for application-only identity.
- **idk** is the **base64**-encoded identity key. This key is returned by an identity resolver that is shared by the client and server components.
- **idp** is the identity provider claim. There are three possible values for this claim: windows, forms, and trusted.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Example server-to-server tokens issued by the application server when calling another server

The following example shows a server-to-server token issued by the application server when requesting access to a resource on another server.

```
{
    iss: 00000003-0000-0ff1-ce00-000000000000@6305dc22-8cb8-4da3-8e76-8d0bbc0499a5
    nameid: 00000003-0000-0ff1-ce00-000000000000@6305dc22-8cb8-4da3-8e76-8d0bbc0499a5
    identityprovider: 00000003-0000-0ff1-ce00-000000000000@6305dc22-8cb8-4da3-8e76-
8d0bbc0499a5
    nbf: 1320176785
    exp: 1320219985
    aud: 00000003-0000-0ff1-ce00-000000000000/mysite.contoso.com@6305dc22-8cb8-4da3-8e76-
8d0bbc0499a5
    trustedfordelegation: true
}
```

The following example shows a server-to-server token issued by an application server when requesting access to a resource on another server. This example includes user identity as an outer token.

```
{
    iss: 00000003-0000-0ff1-ce00-000000000000@6305dc22-8cb8-4da3-8e76-8d0bbc0499a5
    nameid: user@6305dc22-8cb8-4da3-8e76-8d0bbc0499a5
    identityprovider: windows
    nbf: 1320176785
    exp: 1320219985
    aud: 00000003-0000-0ff1-ce00-000000000000/mysite.contoso.com@6305dc22-8cb8-4da3-8e76-
8d0bbc0499a5
    actortoken:
    {
        iss: 00000003-0000-0ff1-ce00-000000000000@6305dc22-8cb8-4da3-8e76-8d0bbc0499a5
        nameid: 00000003-0000-0ff1-ce00-000000000000@6305dc22-8cb8-4da3-8e76-8d0bbc0499a5
        identityprovider: 00000003-0000-0ff1-ce00-000000000000@6305dc22-8cb8-4da3-8e76-
8d0bbc0499a5
        nbf: 1320176785
        exp: 1320219985
        aud: 00000003-0000-0ff1-ce00-000000000000/mysite.contoso.com@6305dc22-8cb8-4da3-8e76-
8d0bbc0499a5
        trustedfordelegation: true
    }
}
```

5 Security

5.1 Security Considerations for Implementers

Security considerations mentioned in the following specifications ought to be considered when implementing this profile:

- Section 10 in The OAuth 2.0 Authorization Protocol [IETFDRAFT-OAuth2.0].
- Section 10 in JSON Web Token (JWT) Specification Draft [IETFDRAFT-JWT-LATEST].
- Security considerations section in OAuth 2.0 Authentication Protocol Extensions [MS-OAUTH2EX].

In addition the following security aspects ought to be considered:

- Access tokens issued by the Security Token Service are **Bearer** tokens and need to be kept confidential in transit and in storage. It is recommended to use a **TLS** (**SSL**) secured channel for transmitting the access tokens.
- Because the augmented user identity information in the outer token is not signed by the application, the receiver of the server-to-server token ought to validate that the value of the **trustedfordelegation** claim is set to true.
- The receiver of the server-to-server token ought to validate that the **aud** (audience) claim in the inner and outer tokens match. Also, it ought to ensure that the token is intended for itself by ensuring that the **aud** claim contains its *hostname*.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft SharePoint Server 2013
- Microsoft SharePoint Foundation 2013
- Microsoft Lync Server 2013
- Microsoft Skype for Business Server 2015
- Microsoft SharePoint Server 2016
- Microsoft Skype for Business Server 2019
- Microsoft SharePoint Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.2: 00000003-0000-0ff1-ce00-000000000000 identifies SharePoint Server 2013.

<2> Section 2.2: 00000002-0000-0ff1-ce00-000000000000 identifies Microsoft Exchange Server 2013.

<3> Section 2.2: 00000004-0000-0ff1-ce00-000000000000 identifies Lync Server 2013.

<4> Section 3.1: Applicable to SharePoint Server 2013.

<5> Section 3.2: Applicable to SharePoint Server 2013.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

R

S

T

V